LOWER BOUNDS FROM COMPLEX FUNCTION THEORY †

Michael Ian Shamos
Departments of Computer Science and Mathematics
Carnegie-Mellon University
Pittsburgh, PA 15213

and

Gideon Yuval ††
Department of Computer Science
The Hebrew University
Jerusalem, Israel

Abstract

We employ elementary results from the theory of several complex variables to obtain a quadratic lower bound on the complexity of computing the mean distance between points in the plane. This problem has 2N inputs and a single output and we show that exactly $N(N-1)/2$ square roots must be computed by any program over +, −, ×, ÷, $\sqrt{\ }$, *log* and comparisons, even allowing an arbitrary field of constants. The argument is based on counting the total number of sheets of the Riemann surface of the analytic continuation to the complex domain of the (real) function computed by any algorithm which solves the problem. While finding an exact answer requires $O(N^2)$ operations, we show that an ε-approximate solution can be obtained in $O(N)$ time for any $\varepsilon > 0$, even if no square roots are permitted.

I. Introduction

The complexity of algebraic problems over the four basic arithmetics has been extensively studied[1]. For the most part, non-linear lower bounds have been very difficult to obtain, requiring the formidable machinery of algebraic geometry[2]. If we allow as primitives such auxiliary functions as *arcsin*, *log*, and *sqrt*, it is widely assumed that proving lower bounds will be even more troublesome, because linear independence arguments no longer apply. The purpose of this paper is to introduce a technique for counting the number of invocations of such functions required during a computation. We will find that, in many cases, multiple-valued functions are easier to count than arithmetics.

Extending the power of {+, −, ×, ÷} by allowing square roots is a temptation that is difficult to avoid. The computational power of these primitives is precisely that of the Euclidean ruler and compass, and the ancient geometers were preoccupied with the question of what could and could not be constructed using them. The matter drew the attention of Gauss, Galois, and Hilbert[3], and there now exists a considerable body of theory relating to square roots. Most of this work, however, pertains to computability, not complexity, and we still have no effective way of counting the number of square roots required to compute functions.

Interestingly, research has been done on determining how many arithmetics are needed to approximate a square root to any given accuracy[4], but there also seems to be some justification for considering square root as a primitive in its own right. First, in algebraic complexity we usually assume straight-line or tree programs with real inputs and infinite-precision operations, so roundoff error is not considered. Also, computer hardware exists in which square roots can be performed as quickly as floating-point multiplications, hence treating the root as a single operation is quite realistic.

The method we shall employ is elementary in concept. Consider a straight-line program over {+, −, ×, $\sqrt{\ }$} (omitting division for the time being), whose input, $X = (x_1, \ldots, x_N)$, is an N-tuple of real numbers. This program computes an analytic function $f(X)$, which may be complex-valued, on the reals. By the identity theorem for analytic functions, there is a unique function $f_c(Z)$, of N complex variables, which equals f when all its arguments are real. This function f_c is said to be the analytic continuation of f. The situation is complicated somewhat by the fact that the square root function is multiple-valued, but it is

†This research was supported in part by the Office of Naval Research under contracts N00014-76-C-0370 and N00014-76-C-0829, and by the National Science Foundation under Grant MCS75-222-55.

††Work performed while this author was a Visiting Scholar at Carnegie-Mellon University.

just this property that will enable us to obtain a lower bound on the number of square roots performed by any program that computes f. The multiple-valuedness of f_c is completely described by its <u>Riemann surface</u>, on which the function is single-valued. The Riemann surface of $\sqrt{z}$ has two sheets, reflecting the fact that square root is bi-valued. The reader who is unfamiliar with these ideas should consult Bieberbach[5] or Knopp[6] for background material. Each square root performed during the execution of our program for f can at most double the number of sheets of the Riemann surface of f_c. Furthermore, all such bifurcations of the Riemann surface must be directly attributable to the square root operations, since the other primitives are unambiguous. It follows that if the Riemann surface of f_c has R sheets, then at least $\log_2 R$ square roots must have been executed by the program that computed f. In this way we will obtain tight lower bounds.

The actual problem that we will use to illustrate the technique is one from computational geometry that has previously resisted solution: Given N points in the plane, N(N-1)/2 pairwise distances are determined. The average of these is a measure of the "spread" of the set and is called the <u>mean intraset distance</u>. This quantity is of interest in pattern recognition, clustering, and multidimensional scaling[7] and we would like to develop a fast algorithm for computing it. The existence of efficient algorithms for calculating related distance measures[8] makes it implausible that this simple function should require $O(N^2)$ operations to compute, but we will show that exactly N(N-1)/2 square roots are required for its evaluation. While this last result is discouraging, particularly if large numbers of points are involved, we explain in a later section how to obtain an ε-approximation to the mean distance in only linear time.

To give the reader a chance to become comfortable with the idea of using complex function theory to analyze real algorithms, we postpone the discussion and begin instead with the simpler, one-dimensional version of the mean distance problem.

II. <u>Mean Distance in One Dimension</u>

Given N points on the line, we can ask a number of questions about the set of distances determined by those points and discuss the complexity of answering our questions. We assume that the points are given as an N-tuple $X = (x_1, \ldots, x_N)$, and consider tree programs over $\{+, -, \times, \div\}$, with binary comparisons allowed. (Square roots are not needed for distance calculations in one dimension.) The distance between points x_i and x_j is given by $|x_j - x_i|$. The maximum distance between any two points, also known as the <u>diameter</u> of the set, can be found in O(N) time by subtracting the smallest x-value from the largest. This can be done in about 3N/2 comparisons[9]. By contrast, the smallest interpoint distance cannot be found in fewer than O(N log N) comparisons[10]. The median distance can be found in O(N log N) time by entirely different methods[11]. In this section we will find upper and lower bounds on computing the <u>mean</u> distance between points on the line.

The mean distance is the sum, S, of the interpoint distances, divided by N(N-1)/2, the number of distinct pairs of points. Since the mean can be computed from S and N in a constant number of operations, we will focus our attention on computing S only. By definition,

$$S = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} |x_j - x_i| = \sum_{i<j} |x_j - x_i| \qquad (1)$$

and it can be computed directly from this expression in $O(N^2)$ operations.

<u>Theorem 1</u>. *The mean distance in one dimension can be found in O(N log N) time, and this is optimal.*

Proof: Let us sort the coordinates in O(N log N) time so that x_i is now the i'th smallest value. The interval between any pair of points is made up of intervals between consecutive points. In the sum S, the distance between x_i and x_{i+1} is counted i(N-i) times, once for every pair of the form (u,v), $1 \le u \le i$, $i+1 \le v \le N$, of which there are i(N-i). Thus, after the sort,

$$S = \sum_{i=1}^{N-1} i(N-i)\,(x_{i+1} - x_i) \qquad (2)$$

which may be rewritten as

$$S = \sum_{i=1}^{N} (2i - N - 1)\, x_i \quad . \qquad (3)$$

Equation (3) shows that, once the sort is accomplished, S can be computed in linear time.

To prove the lower bound, we show that any tree program which computes S can be made to sort the x's with no additional comparisons. Consider the partial derivative of S with respect to any one of the inputs:

$$\frac{\partial S}{\partial x_i} = 2i - N - 1 \quad . \qquad (4)$$

From this partial derivative the rank of x_i can be determined. Since, within a given branch of the computation tree, the partials can be evaluated by arithmetics alone, O(N log N) comparisons must have been made. □

III. Mean Distance in Two Dimensions

The effect of dimension on complexity in geometric problems has only begun to be studied, but we know that the maximum[12] and minimum[10] interpoint distances in the plane can be found in O(N log N) time. The best known algorithm for computing the median distance in the plane is quadratic, although no better lower bound than O(N log N) has been shown. We can find the mean distance from the sum

$$S_2 = \sum_{i<j} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad . \qquad (5)$$

Let $M = N(N-1)/2$. It may appear that since there are only 2N parameters in equation (5), not all of the M square roots are algebraically independent and possibly some may be eliminated. Our job is to show that, in fact, (5) is optimal with respect to square roots.

The square root function is multiple-valued: the square root of 9 is both 3 and -3. This behavior is clarified in complex analysis by introducing an elegant device known as the Riemann surface, which reflects the fact that the square root of a number z in the complex plane is not a function of z alone, but also of the path that was taken to arrive at z. Beginning at z, let us go out and return via a path that encircles the origin. During this trip, the square root will have changed sign. This is a consequence of the relation $\sqrt{z} = \sqrt{r}\,\exp(i\theta/2)$, where (r,θ) are the polar coordinates of z. This means that if we replace θ by $\theta + 2\pi$, the square root will change by a factor of $\exp(2\pi i/2) = -1$. We express this by saying that the Riemann surface of $\sqrt{z}$ consists of two sheets that cover the complex plane and, in traversing a path that surrounds the origin, we move from one sheet to the other. The "ambiguity" of a function is completely characterized by its Riemann surface.

Now let us consider complex functions of N complex variables, defined by finite expressions in those variables whose only multiple-valued primitive is $\sqrt{\ }$. Let the Riemann surface of such a function f have R sheets. Since each square root introduces at most a twofold ambiguity, at least $\log_2 R$ square roots must have been used to define f. In order to use this to prove lower bounds, we must establish a connection between a straight-line program over the reals and the Riemann surface of some function of many complex variables. We must further be able to count the number of sheets of that surface.

A complex analytic function of one variable may be defined completely by giving its value on any finite interval, no matter how small. This is known as the Identity Theorem[13] or Principle of Analytic Continuation. Given a real function, defined on an interval of the real line, there is at most one complex analytic function, defined on the whole plane, which agrees with the given real function on the given interval. In the case of square root, this means that we cannot get rid of multiple-valuedness by trying to restrict the domain of definition. Suppose we unabashedly declare that what we really mean by $\sqrt{x}$ is $|\sqrt{x}|$. This is to no avail, because the analytic continuation of $|\sqrt{x}|$, as well as of $\sqrt{x}$, is $\sqrt{z}$ again. Before proceeding further, we must quote a version of the Identity Theorem for several complex variables[14]:

> "If f_1 is analytic in a domain D_1, and f_2 in a domain D_2, if $D_1 \cap D_2$ is a non-empty domain, and if $f_1(Z)$, $f_2(Z)$ have equal values in a real environment of a point of $D_1 \cap D_2$, then $f_1(Z)$ and $f_2(Z)$ are analytic continuations of each other; i.e. there exists a unique function $f(Z)$, analytic in $D_1 \cup D_2$, which coincides with f_1 in D_1 and f_2 in D_2."

We are now ready to show how a unique complex function corresponds to any program which computes a real analytic function.

Consider a straight-line program over $\{+, -, \times, \sqrt{\ }\}$ with N real inputs, which computes a real function $f(X) = f(x_1, \ldots, x_N)$. This function is analytic, except for square root singularities (which are isolated), on the real hyperplanes. By the Identity Theorem, if we take D_1 and D_2 to be the singularity-free regions of the real hyperplanes, there is a unique function f_c, of N complex variables, which coincides with f on the reals. If f_c is multiple-valued, this can only arise from the presence of square roots in the program that defines f.

It is important to realize that the mere presence of k explicit square roots in the program defining f does not guarantee that the Riemann surface of f_c will have 2^k sheets. For example, the surface of $\sqrt{3z} + \sqrt{7z}$ has two sheets, not four. This is true because the two roots are not "independent" in the sense that there is no closed path in the complex plane that reverses the sign of one without also reversing the sign of the other. The ambiguity is thus two-fold, not four-fold.

The last complication remaining before we can prove the main theorem concerns disconnected sheets. The Riemann surface of $g(z) = \sqrt{z^2}$ has two sheets, one corresponding to the function $g(z) = +z$, the other to the function $g(z) = -z$. There is, however, no path leading from one sheet to the other, so once we choose a sheet, the function is completely unambiguous. Thus we are not surprised that $g(z) = \sqrt{z^2}$ can be computed without square roots. The contention that $\log_2 R$ square roots are required to evaluate a function whose Riemann surface has R sheets is valid if and only if there is a path on the Riemann surface that touches all R sheets.

Theorem 2. *Any straight-line program over* $\{+, -, \times, \sqrt{\ }\}$ *which computes the mean distance between* N *points in the plane must perform at least* $N(N-1)/2$ *square roots.*

Proof: From the above discussion, we need only show that the Riemann surface of the analytic continuation of the sum in equation (5) has 2^M sheets, where $M = N(N-1)/2$, and that these sheets are connected. It suffices to show that each of the terms in (5) becomes singular at some point. A closed path taken around that point will cause one root only to change sign; the others will remain unaffected. (This will prove that none of the ambiguities degenerate, as they do in the case of $\sqrt{3z} + \sqrt{7z}$.) To show how this may be done, we consider the distance between points 1 and 2. Take $x_k = y_k = k$, $k > 1$, and $x_1 = 1$, $y_1 = 2 + i$. Then $(x_1 - x_2)^2 + (y_1 - y_2)^2$ is zero, and its root is singular. None of the other interpoint distances are zero. □

We will now enhance the power of Theorem 2 by discussing a number of generalizations of the straight-line model. A refinement of the argument can be made to show that allowing divisions does not reduce the number of square roots required. The Identity Theorem does not apply immediately because a program that uses divisions computes a meromorphic function, not an analytic one. We can still perform the analytic continuation, however, if we do not try to continue through a pole.

An immediate consequence of the Riemann surface technique is that no number of single-valued analytic function evaluations can reduce, even by one, the number of square roots required by equation (5). We can thus add *sin(x)*, *cos(x)*, *Γ(x)*, *exp(x)*, or any other entire function to the set of primitives allowed in Theorem 2. Even other multiple-valued functions may not help. While higher-order roots contribute many sheets to the Riemann surface, these sheets have the wrong structure. Suppose we allow fourth-roots. While we may square a fourth-root to obtain a square root, no unsquared fourth-root is of any use because we must traverse a path around its singularity four times to return to the root's original value, not twice, as for square roots. Likewise, even though the Riemann surface of *log(z)* has an infinite number of sheets, these also have the wrong structure. Note that if both *exp* and *log* are allowed, we can express a square root as $\sqrt{x} = \exp(\frac{1}{2}\log x)$, but this has the effect of replacing a single square root by two other operations, and cannot serve to reduce the total number of steps required.

Finally, we demonstrate that allowing comparisons will not enable us to eliminate any square roots. The input to our algorithm is a real 2N-tuple, which may be viewed as a point in Euclidean 2N-space. If the maximum number of comparisons performed during any execution of the program is c, then real 2N-space is partitioned into at most 2^c equivalence classes, within which the outcomes of the decisions are identical. Each equivalence class defines a subset of 2N-space. Since the number of subsets is finite, one of them must contain an interval, and the required analytic continuation can be performed. The fact that comparisons do not help should not have been unexpected: they simply do not contribute to multiple-valuedness.

We summarize all the above results as

Theorem 3. *Any comparison-tree program using arithmetics, entire functions, k-th roots, and logarithms, which computes the mean intraset distance in the plane, must use* $O(N^2)$ *steps in the worst case.* □

IV. Approximations to the Mean Distance

Having just derived a quadratic lower bound for computing the mean intraset distance in the plane, we will show how it can be circumvented. We say that S^* is an ε-approximation to S if, for $\varepsilon > 0$, we have

$$\left| \frac{S^* - S}{S} \right| < \varepsilon \quad . \tag{6}$$

Theorem 4. *An* ε*-approximation to the mean distance in one dimension can be found in* $O(N \log(1/\varepsilon))$ *time.*

Proof: We give an algorithm. Let $k = 1 + 2/\varepsilon$ and find the k-tiles that partition the x_i into k groups of approximately equal size. This can be done in $O(N \log k)$ time by recursive application of linear selection[11 15]. These partitioning elements divide the x_i into k buckets and, in an additional $O(N \log k)$ time by binary insertion, we can determine the bucket into which each x_i falls. Let c_j be the centroid of bucket j, $j = 1, \ldots, k$. All of the c_j can be computed in linear time and are already in sorted order. We now approximate by assuming that all of the points in a group lie at the centroid of that group and compute the mean distance under that assumption using equation (3). This can be done in linear time. By this method, all distances between pairs of points lying in different groups are counted exactly; all distances between points lying within a group are ignored. However, the sum of distances between points in group j is less than or equal to the sum of distances between pairs of points, one lying in group j and the other lying in any different group i. Thus, of the $k(k-1)/2$ inter-group distances, for each of the k distances ignored, $(k-1)/2$ are computed exactly, so the actual relative error is not greater than $2k/k(k-1) = 2/(k-1) = \varepsilon$. □

Theorem 5. *An ε-approximation to the mean distance in the plane can be found in* $O\left(N \frac{\log \frac{1}{\varepsilon}}{\sqrt{\varepsilon}}\right)$ *time.*

Proof: We show that an ε-approximate algorithm in two dimensions can be obtained by solving $\sim 1/\sqrt{\varepsilon}$ approximate problems in one dimension. Consider the metric in the plane whose unit ball is a regular $2m$-gon inscribed in the unit circle. The distance between two points in this metric is the normalized sum of their distances projected on m different axes. (The rectilinear, or Manhattan, metric corresponds to the case $m = 2$.) When all N points are projected onto a line, we may compute an ε-approximation to their mean distance on that axis in $O(N \log(1/\varepsilon))$ time by Theorem 4. Doing this over all m axes requires $O(mN \log(1/\varepsilon))$ time. The relative error is bounded by the ratio of the minimum and maximum distance from the origin to a point on the unit ball of the metric. For a regular $2m$-gon this ratio is given by

$$1 - \cos \frac{\pi}{2m} \sim \frac{\pi^2}{8m^2} \quad . \qquad (7)$$

The error may be reduced by a factor of two if we use the metric whose unit ball is the polygon lying midway between the regular inscribed and circumscribed $2m$-gons. We then have an error less than $1/m^2$ and it suffices to choose $m = 1/\sqrt{\varepsilon}$. □

The metric introduced above can be used to provide fast approximations for a variety of geometry problems.

V. Discussion

The reader may have the feeling that something suspicious has just taken place. After all, what do complex functions have to do with algorithms over the reals ? To this we can only reply that the principle of analytic continuation prevents the building of fences that restrict well-behaved functions to the real domain. It may be comforting to recall the assistance that the theory of residues provides in evaluating real definite integrals. Another reader may become squeamish at the mention of the theory of many complex variables; he will be reassured to hear that Theorem 2 can be proven by carrying out the analytic continuation in each variable separately, so that no results more sophisticated than the Identity Theorem for one variable need be applied. For the studious reader, we supply a homework problem:

Equation (1) for the mean distance in one dimension can be rewritten in terms of square roots as

$$S = \sum_{i<j} |x_j - x_i| = \sum_{i<j} \sqrt{(x_j - x_i)^2} \quad . \qquad (8)$$

Why doesn't Theorem 2 imply a quadratic lower bound on the complexity of evaluating (8) ? The reader who understands the answer to this question will have grasped the essence of the argument.

In retrospect, our methods have much of the flavor of independence techniques. We show that the square roots are "independent" because they can each be made singular separately. By identical means we can construct problems of arbitrary polynomial complexity.

All the fruitful uses of imaginaries, in Geometry, are those which begin and end with real quantities, and use imaginaries only for the intermediate steps.

Bertrand Russell[16]

VI. Summary

We have shown how to use techniques from complex function theory to obtain lower bounds for comparison tree programs whose primitives have been augmented to include such multiple-valued functions as $\sqrt{\ }$, *log*, and higher-order roots. In certain cases, functions that are analytic everywhere, such as *sin* and *exp*, are of no help in reducing the number of square roots required to evaluate a function. By counting the number of sheets of the Riemann surface of a function determined by the algorithm in question, we are able to show that a quadratic number of operations are needed to compute the mean distance between points in the plane. At the same time, we exhibit efficient approximation algorithms for this problem.

Acknowledgements

We wish to thank Dave Jefferson and Joe Traub for carefully picking apart our arguments and thereby helping to clarify the presentation.

References

1 Borodin, A. and Munro, I. The Computational Complexity of Algebraic and Numeric Problems. American Elsevier (1975), 174 pp.

2 Strassen, V. *Die Berechnungskomplexetät von elementarysymmetrischen Funktionen und von Interpolationskoeffizienten.* Numerische Mathematik 20,3 (1973), 238-251.

3 Hilbert, D. Foundations of Geometry. (1899). Edited and reprinted by Open Court Press, 1971.

4 Kung, H.T. *A Bound on the Multiplicative Efficiency of Iteration.* JCSS 7,4 (1973), 334-342.

5 Bieberbach, L. Conformal Mapping. Chelsea (1953).

6 Knopp, K. Theory of Functions, Part II. Dover (1947), 150 pp.

7 Duda, R.O. and Hart, P.E. Pattern Classification and Scene Analysis. Wiley (1973), 482 pp.

8 Shamos, M.I. Problems in Computational Geometry. Springer-Verlag, to appear.

9 Pohl, I. *A Sorting Problem and its Complexity.* CACM 15,6 (1972), 462-464.

10 Shamos, M.I. and Hoey, D. *Closest-Point Problems.* Proc. Sixteenth Annual IEEE Symposium on Foundations of Computing. Berkeley, CA (1975), 151-162.

11 Shamos, M.I. *Geometry and Statistics: Problems at the Interface.* Proc. Symposium on New Directions and Recent Results in Algorithms and Complexity. Academic Press (1977).

12 Shamos, M.I. *Geometric Complexity.* Proc. Seventh Annual ACM Symp. on Theory of Computing. (1975), 224-233.

13 Marsden, J.E. Basic Complex Analysis. Freeman (1973), 472 pp.

14 Bochner, S. and Martin, W.T. Several Complex Variables. Princeton (1948).

15 Yao, A.C.-C. *An $O(|E| \log \log |V|)$ Algorithm for Finding Minimum Spanning Trees.* Information Processing Letters 4(1975), 21-23.

16 Russell, B. Foundations of Geometry, p.45. Cambridge (1897).